Using State Abstractions to Compute Personalized Contrastive Explanations for AI Agent Behavior

Sarath Sreedharan, Siddharth Srivastava, Subbarao Kambhampati

CIDSE, Arizona State University, Tempe, AZ 85281 USA
{ *ssreedh3, siddharths, rao* } @ *asu.edu*

Abstract

There is a growing interest within the AI research community in developing autonomous systems capable of explaining their behavior to users. However, the problem of computing explanations for users of different levels of expertise has received little research attention. We propose an approach for addressing this problem by representing the user's understanding of the task as an abstraction of the domain model that the planner uses. We present algorithms for generating minimal explanations in cases where this abstract human model is not known. We reduce the problem of generating an explanation to a search over the space of abstract models and show that while the complete problem is NP-hard, a greedy algorithm can provide good approximations of the optimal solution. We empirically show that our approach can efficiently compute explanations for a variety of problems and also perform user studies to test the utility of abstractions in explanations.

Keywords: Explanations for Plans, Abstractions, Contrastive Explanations

1. Introduction

AI systems have the potential to transform society by assisting humans in diverse situations ranging from extraplanetary exploration to assisted living. In order to achieve this potential, however, humans working with such systems need to be able to understand them just as they would understand human team members. This presents a number of challenges because most humans do not understand AI algorithms and their behavior at the same intuitive level that they understand other humans. Handling and possibly overcoming such knowledge asymmetry requires us to develop and deploy AI systems that are capable of providing cogent explanations for their actions/decisions to end users. A significant challenge for any such system would be the fact that more often than not, the AI system may be modeling and reasoning about the task with much greater fidelity than the user is aware of (or capable of reasoning with). While there have been a number of recent works on the problem of explaining plans and actions chosen by agents (readers can refer to the survey [1] for previous works in this direction), they have generally assumed that the user understands the task at the same level of abstraction as the agent in question.

In this paper, we propose a new approach to this problem where the agent explains its ongoing or planned behavior in a way that is both tailored to the user's background and is designed to reduce cognitive burden on the user's end. This is done by modeling a user's expertise, or the level of detail at which a user understands the task using abstracted models. We can estimate this level based on questions that the user asks and provide explanations that are close to this estimated level of expertise.

We consider explanations in the framework of counterfactual reasoning, where a user who is confused by the agent's activity (or proposed activity) presents alternative behavior that they would have expected the agent to execute. This aligns with the widely held belief that humans expect explanations to be *contrastive* [2]. In keeping with the terminology used in social sciences literature, we will denote the set of alternative behaviors as *foils* to the proposed robot behavior.

For instance, consider a mission-control operator who needs to manage an autonomous robot on Mars in the midst of a sandstorm that could present valuable data for analysis. If the robot proposes to go back to the base before going to a vantage point for observing the storm, the operator would naturally be perplexed, and may be motivated to ask the rover why it didn't go directly to the vantage point. While answering the operator's queries it is important that the explanation being given is tailored to meet the user's background knowledge. Here an explanation that informs the operator of some specific mission goals that warranted this unintuitive plan, for example *"I am required to drop the collected samples in the base before going to the vantage point"*, may be preferred over a detailed explanation involving the specifics of the battery model or the rover motors (for example *"Motors of model #2310 needs to be recalibrated after every 20 miles and I need to go to base to recaliberate"*). As far as the rover is concerned, both of these explanations are equally valid reasons to choose the circuitous route, but a mission control operator may find the former easier to understand while an engineer may better appreciate the latter. This level of user specificity requires methods that estimate possible models that can capture the user's level of understanding of the task. As mentioned, we will make use of the questions (i.e the foils) raised by the user for the specific task at hand (and potentially even the history of previous interactions) to build such estimates. Accurate estimate of the user's expertise not only lets us control the level of detail in explanation but also allows us to provide the most concise explanation (by avoiding unnecessary details) and thereby reduce the cognitive burden of the user.

In this paper we present the **Hierarchical Expertise-Level Modeling** or the **HELM** approach for facilitating such context and user-specific explanations. We assume that the user's understanding of the task is an abstraction of the model used by the robot; which captures both the limited information and computational capabilities of the user. HELM generates appropriate explanations by searching through a *model lattice* of possible abstractions of the agent's model. The model lattice provides a concise way for the system designer to encode their prior knowledge about potential users. Each model within this lattice represents a different level of understanding of the task, with the highest fidelity representation (corresponding to the most detailed understanding of the domain used by the robot) forming the base of the lattice and the model representing the most naive understanding of the task (for example one held by a lay person) forming the highest nodes. Since the user's level of expertise is unknown to the agent, it has to estimate the human model before searching for an explanation.

We focus on contrastive explanations, where an explanation that is an answer to a question of the form “Why P and not Q?”, in our case, P and Q are stand-ins for the current robot plan and the foil respectively. Most existing works in explanation for plans have focused on answering the first part “Why P?” (for example works like [3, 4] have looked at identifying causal explanations for each action), so the majority of this work will focus on finding a concise explanation for the latter part of the full question. Specifically, our explanations will consist of model information that may be absent in the user’s abstract model and possible proofs for foil failure. Thus, in addition to helping convince the user of the incorrectness of the foils in question, the explanations should also shift the user’s model to a more accurate model in the lattice. This approach could be understood as a variant of the model refinement methods discussed in the counter-example guided model checking (CEGAR) literature [5]. Our methods extend these principles to settings with uncertainty regarding the current level of abstraction of the model (a non-issue in the model-checking settings where CEGAR methods are typically used).

This paper generalizes and extends our recent work [6] with extended theoretical and empirical results and exposition. In addition to clarifying the concepts our contributions include

- We consider the use of non-standard lattices as a way to allow designers to incorporate more information about the user’s model in to the explanation generation process and discuss potential computational tradeoff introduced by the use of such lattice types over the ones considered in the rest of the paper.

- We investigate the use of such methods for domains that contain state dependent costs (hence affected by the abstraction) and discuss the potential explanatory dialogue that could occur in such settings.

- We also show how our method could be used in cases where the user

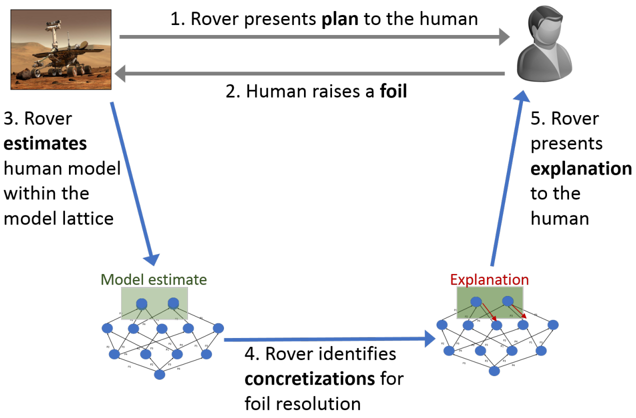

1. Rover presents **plan** to the human
2. Human raises a **foil**
3. Rover **estimates** human model within the model lattice
Model estimate
4. Rover identifies **concretizations** for foil resolution
Explanation
5. Rover presents **explanation** to the human

Figure 1: An illustration of the hierarchical explanation process. The human observer who views the task at a higher level of abstraction expects the rover to execute a different plan from the one chosen by the rover. The rover presents the human with an explanation it believes will help resolve the foils in the human's updated model.

model may not just be abstract but the user may also hold erroneous beliefs about the task.

- We perform a user study to verify the utility of abstraction in generating explanation that are easier for users to work with.

The rest of this paper is structured as follows. Section 2 a brief overview of the background and in Section 3 we present our formal framework. Section 4 covers different approaches for generating explanations and sections 4.1, 4.2 and 4.3 extend these methods to more general settings. Section 5 presents evaluations of the method. In sections 6 and 7, we will discuss the related work and possible future directions.

2. Background

In this work, we focus on abstractions that form models by projecting out state fluents. While the presentation in the following sections is equally valid for both predicate and propositional abstractions, we will focus on propositional abstractions to keep our formulation clear and concise and later discuss potential changes required to meet the requirements of predicate abstractions. We will look at planning models of the form $\mathcal{M} = \langle P, S, A, I, G \rangle$ where P gives the set of state fluents, S the set of possible states, A the set of actions, I the initial state and G the goal. Each state $s \in S$ is uniquely represented by the set of propositions that are true in that state, i.e, $s \subseteq P$.

Each action $a \in A$ is associated with a set of positive preconditions prec_a^+ (specified as a conjunction of propositions) and negative preconditions prec_a^- that need to hold for the effects (e_a) of that action to be applied to a particular state. Each effect set e_a can be further separated into a set of add effects e_a^+ and a set of delete effects e_a^-. The result of executing an action a on a state s in this setting is defined as $a(s) = (s \cup e_a^+) \setminus e_a^-$, if $\text{prec}_a^- \subseteq s \ \wedge \ \text{prec}_a^- \cap s = \emptyset$. A plan π is defined as a sequence of actions ($\langle a_1, .., a_n \rangle$, n being the size of the plan), and a plan is said to solve $\mathcal{M}$ (i.e, $\pi(I) \models_{\mathcal{M}} G$) if $\pi(I) \supseteq G$.

Automated planning has a long tradition of employing abstraction both for plan generation (cf. [7]) and for generating heuristics (cf. [8, 9]) and a number of different abstraction schemes have been proposed in these works. In fact, state abstractions as presented in this work have been widely used in pattern databases and are referred to as projections in that literature (cf. [10, 11]). Following works like [8, 12], we will also use the concept of a transition system induced by the planning model to define state abstractions. Intuitively, a transition system constitutes a graph where the nodes represent possible states, and the edges capture the transitions between the states that are valid in the corresponding planning model.

Formally a transition system $\mathcal{T}$ corresponding to a model $\mathcal{M}$ can be represented by a tuple of the form $\mathcal{T} = \langle S, L, T, s_o, S_g \rangle$, where S is the set of possible states in $\mathcal{M}$, L is the set of transition labels (corresponding to the action that induce that transition), T is the set of possible labeled transitions, s_0 is the initial state and S_g is the set of states that satisfies the goal specified by $\mathcal{M}$. We will refer to $\mathcal{T}$ to be the *safe transition* system induced by a model $\mathcal{M}$, if and only if, for any labeled transition $\langle s, a, s' \rangle \in T$, we have $\text{prec}_a^+ \subseteq s$ and

$\text{prec}_a^- \cap s = \emptyset$. Through most of this work we will focus our attention on cases where the semantics of the planning task is defined in terms of safe transition systems.

Definition 1. A propositional abstraction function f_Λ for a set of propositions Λ and state space S, defines a surjective mapping of the form $f_\Lambda : S \rightarrow X$, where X is a projection of S, such that for every state $s \in S$, there exists a state $f_\Lambda(s) \in X$ where $f_\Lambda(s) = s \setminus \Lambda$.

Definition 2. For a planning model $\mathcal{M} = \langle P, S, A, I, G \rangle$ with a corresponding transition system $\mathcal{T}$, a model $\mathcal{M}' = \langle P', S', A', I', G' \rangle$ with a transition system $\mathcal{T}'$ is considered an **abstraction of $\boldsymbol{\mathcal{M}}$** for a set of propositions Λ, if for every transition $s_1 \xrightarrow{a} s_2$ in $\mathcal{T}$ corresponding to an action a, there exists an equivalent transition $f_\Lambda(s_1) \xrightarrow{a'} f_\Lambda(s_2)$ in $\mathcal{T}'$, where a' is part of the new action set A'.

We will slightly abuse notation and extend the abstraction functions to models and actions, i.e in the above case, we will have $\mathcal{M}' \in f_\Lambda(\mathcal{M})$ (where $f_\Lambda(\mathcal{M})$ is the set of all models that satisfy the above definition for the set of fluents Λ) and similarly we will have $a' \in f_\Lambda(a)$. As per Definition 2, the abstract model is *complete* in the sense that all plans that were valid in the original model will have an equivalent plan in this new model. We will use the operator $\sqsubset$ to capture the fact that the model $\mathcal{M}'$ is an abstraction of $\mathcal{M}$, i.e if $\mathcal{M} \sqsubset \mathcal{M}'$ then there exist a set of propositions Λ such that $\mathcal{M}' \in f_\Lambda(\mathcal{M})$.

2.1. Designing Complete Abstractions

While there exists a number of works that have looked at the problem of designing abstractions (cf. [13, 7, 12]), unfortunately many of these works have considered directly updating transition system or using specialized or more expressive problem formulation to capture abstract models. Thankfully, the fact that we are interested in complete abstractions (as opposed to sound abstractions) means we can employ simpler model transformation schemes to generate abstract models. In particular, we will consider transformations that simply drops the set of literals to be abstracted from all the action definitions, i.e,

Theorem 1. *For a given model $\mathcal{M} = \langle P, S, A, I, G\rangle$ and a set of propositions Λ, a model $\mathcal{M}' = \langle P', S', A', I', G'\rangle$ is a complete abstraction under safe execution semantics for Λ, if $P' = P - \Lambda$, $S' = [S]_{f_\Lambda}$, $I' = f_\Lambda(I)$, $G' = G \setminus \Lambda$ and for every $a \in A$ (where $a = \langle prec_a^+, prec_a^-, eff_a^+, eff_a^-\rangle$) there exists $a' \in A'$, such that $a' = \langle prec_a^+ \setminus \Lambda, prec_a^- \setminus \Lambda, eff_a^+ \setminus \Lambda, eff_a^- \setminus \Lambda\rangle$.*

Proof Sketch. To see why the new model would be an complete abstraction, consider a transition $\langle s, a, s'\rangle$ induced by $\mathcal{M}$. Now as per the definitions of safe transition systems, we know that $s \subseteq \text{prec}_a^+$ and $s \cap \text{prec}_a^- = \emptyset$ and $s' = s \setminus \text{eff}_a^- \cup \text{eff}_a^+$. Its easy to see that given this setting, $(s \setminus \Lambda) \subseteq (\text{prec}_a^+ \setminus \Lambda)$ and $(s \setminus \Lambda) \cap (\text{prec}_a^- \setminus \Lambda) = \emptyset$, which means there must be an action $a' \in A'$ that is executable in $f_\Lambda(s)$. Similarly we can show the result of executing a' must be $f_\Lambda(s)$, this shows that $\mathcal{M}'$ is a complete abstraction of $\mathcal{M}$ as every transition induced by it is present in the transition system induced by $\mathcal{M}'$. □

An important point to note here is that this transformation scheme generates a unique abstract model for each model and proposition set, and we will denote this unique model as $f_\Lambda(\mathcal{M})$. For the rest of the paper, we will mainly focus on this method to induce the abstractions, but general framework of explanation generation discussed in this paper can be adapted to other methods of generating abstract models. In cases, where we prove specific results or present optimization that rely on this abstraction procedure we will denote the abstraction function by f_Λ^{safe} to differentiate it from other methods. With the definition of abstraction and related notations in place, we will look at our explanatory setting and a way to capture the space of possible user models that would allow for efficient estimation of unknown user model given user queries. While the above operation is defined for propositional fluents, we can perform similar operations on the lifted domain, where projecting out a predicate would correspond to projecting out a set of propositional fluents from the grounded domain.

3. Hierarchical Expertise-Level Modeling

As mentioned earlier, we are investigating explanatory settings where the user's understanding of the task can be represented as an abstraction of the robot's model. While the exact level of abstraction may be unknown, given a set of candidate state fluents that may be missing from the human model, we can capture the potential models and their relationship through a **model lattice**

Definition 3. For a model $\mathcal{M}^{\#}$, the *model lattice* $\boldsymbol{\mathcal{L}}$ is a tuple of the form $\mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle$, where $\mathbb{M}$ is the set of lattice nodes such that $\mathcal{M}^{\#} \in \mathbb{M}$ and $\forall\ \mathcal{M}' \in \mathbb{M}, \mathcal{M}^{\#} \sqsubseteq \mathcal{M}'$, $\mathbb{E}$ is the lattice edges, $\mathbb{P}$ is the superset of propositions considered for abstraction within this lattice and ℓ is a function mapping edges to labels. Additionally, for each edge $e_i = (\mathcal{M}_i, \mathcal{M}_j)$ there exists a proposition $p \in \mathbb{P}$ such that $f_p(\mathcal{M}_i) = \mathcal{M}_j$ and $\ell(\mathcal{M}_i, \mathcal{M}_j) = p$.

Thus each edge in this lattice corresponds to an abstraction formed by projecting out a single proposition (represented by the label of the edge). We can also define a concretization function γ_p that retrieves the model that was used to generate the given abstract model by projecting out the proposition p, i.e, $\gamma_p(\mathcal{M}) = \mathcal{M}'$ if $(\mathcal{M}', \mathcal{M}) \in \mathbb{E}$ and $\ell(\mathcal{M}', \mathcal{M}) = p$ else $\gamma_p(\mathcal{M}) = \mathcal{M}$.

For a given lattice, if each node in $\mathbb{M}$ has an incoming edge for every proposition missing from its corresponding model then we will refer to such lattices as being *Proposition Conserving* lattices.

Definition 4. *A lattice $\mathcal{L}$ is proposition conserving-iff for any model $\mathcal{M} \in \mathbb{M}$ ($\mathcal{M} = \langle P_{\mathcal{M}}, S_{\mathcal{M}}, A_{\mathcal{M}}, I_{\mathcal{M}}, G_{\mathcal{M}} \rangle$) and $\forall p \in \mathbb{P}$, if p is not in $P_{\mathcal{M}}$ then there exists a model $\mathcal{M}' \in \mathbb{M}$, such that $(\mathcal{M}', \mathcal{M}) \in \mathbb{E}$ and $\ell(\mathcal{M}', \mathcal{M}) = p$).*

Notice that enforcing conservation of propositions doesn't require any further assumptions about the human model and can be easily ensured while generating the lattice. Additionally, we will call a proposition conserving lattice that contains an abstract node corresponding to each possible subset of $\mathbb{P}$ as the

Complete Abstraction Lattice for $\mathcal{M}$ given $\mathbb{P}$. The earlier parts of this paper will assume a proposition conserving lattices as they will allow us to simplify discussions and provide efficient solutions. In later sections, we will relax these assumptions and will look at potential tradeoffs for using non-proposition conserving lattices.

We also assume that all abstraction functions used in generating the models in the lattice are commutative and idempotent, i.e., $f_{p_2}(f_{p_1}(\mathcal{M})) = f_{p_1}(f_{p_2}(\mathcal{M}))$ and $f_{p_1}(f_{p_1}(\mathcal{M})) = f_{p_1}(\mathcal{M})$. In the wider literature, a lattice is generally defined to have a unique maximal element and a unique minimal element. While the abstraction lattices we consider in this work will have a unique minimal element (i.e the most concrete nodes), we do not assume that the lattices have a single maximal node (Figure 2 presents an example lattice that does not have a unique maximal node), in that sense the abstraction lattice may be better understood as meet-semilattices, but we will use the term model lattice or abstraction lattice for convenience.

As mentioned earlier, we consider an explanation generation setting where the human observer (H) uses a task model (this model will be denoted as $\mathcal{M}_H = \langle P_H, S_H, A_H, I_H, G_H \rangle$), that is a more abstract version of the robot's model ($\mathcal{M}_R = \langle P_H, S_R, A_R, I_R, G_R \rangle$). While the robot may not know $\mathcal{M}_H$, it knows that $\mathcal{M}_H$ is a member of the set $\mathbb{M}$ for the lattice $\mathcal{L}$. The human comes up with a **foil set F** $= \{\pi_1, \pi_2, ..., \pi_m\}$ that the robot needs to refute by providing an explanation regarding the task. The explanation should contain information about specific domain properties (i.e., state fluents) that are missing from the human's model, how these properties affect different actions (For example, which actions use these propositions as preconditions and which ones generate/delete them) and how the inclusion of these fluents result in the invalidity of the given foils. To illustrate the utility of such explanations consider an example involving a simplified version of the rover domain mentioned earlier.

Example 1. *Let us suppose that the rover uses a modified version of the IPC rover domain [14] that also takes into account the battery level of the rover. Each* rover operation has a different energy requirement, and the battery level needs to be above a predefined threshold for it to execute them, e.g., it can perform rock sampling only if the battery level is above 75%. Furthermore, the rover needs to visit the base station (i.e., the lander) and perform a reset action to recharge its batteries.

The rover knows that the human observer is at most ignorant of its energy requirements, ability to use solar cells and/or storage capabilities. So the model lattice $\mathcal{L}$ needs to consider abstractions corresponding to the following propositions

$\mathbb{P}$={battery_level_above_25_perc, battery_level_above_50_perc, battery_level_above_75_perc, full_store1, solar_panels_activated}.

Figure 2 shows the lattice that the robot would use in this setting. Here we will create each abstract model by following the process discussed in section 2.1. For example, consider the action sample_rock_store0_w1, it has the following definition

$$\langle\{\textsf{battery\_level\_above\_75\_perc},\ \textsf{at\_w1},\ \textsf{empty\_store1},\ \textsf{has\_store\_store1}\},\{\},$$
$$\{\textsf{full\_store1},\ \textsf{has\_rock\_sample}\},\{\textsf{empty\_store1},\ \textsf{battery\_level\_above\_75\_perc}\}\}\rangle$$

Now in an abstract version of this model, if the propositions full_store1, battery_level_above_75_perc are dropped the definition becomes

$$\langle\{\textsf{at\_w1},\ \textsf{has\_store\_store1}\},\{\},$$
$$\{\textsf{has\_rock\_sample}\},\{\textsf{empty\_store1}\}\}\rangle$$

Here the robot presents the plan

$$\pi_R = \langle\ \textsf{navigate\_w0\_lander},\ \textsf{reset\_at\_lander},$$
$$\textsf{navigate\_lander\_w1},\ \textsf{sample\_rock\_store0\_w1}\rangle$$

and a naive observer may respond by proposing the foil set with a single plan

$$F = \{\langle\ \textsf{navigate\_w0\_w1},\ \textsf{sample\_rock\_store0\_w1}\ \rangle\}$$

If the observer was an engineer, they might instead raise a foil that already takes into account the energy requirements

$$F' = \{\langle \textsf{\textit{navigate\_w0\_w1}}, \textsf{\textit{receive\_energy\_from\_solar\_cells}}, \textsf{\textit{sample\_rock\_store0\_w1}} \rangle\}$$

If the robot knew that the human was ignorant about all the battery level predicates and nothing else, the robot could help resolve the naive foil by informing them about the fact that action sample rock requires the battery to be above 75% (i.e describing the proposition battery_level_above_75_perc). In terms of the human model, this would involve setting the value of the proposition battery_level_above_75_perc false in the initial state, updating the precondition of sample_rock_store0_w1 to include the fact (among other actions) and adding it as an add effect to the action reset_at_lander. In this updated model the human foil can no longer achieve the goal. In the case, of expert foil, the robot would need to inform the user about the proposition solar_panels_activated and that the action receive_energy_from_solar_cells require the solar panels to be activated which is not true for the rover. Thus in each case explanations to be provided to user can be generated once we know the set of propositions whose concretization is required to refute the given foils (henceforth referred to as *explanatory fluent set*).

Definition 5. *Let $E = \{p_1, ..., p_n\}$ be a set of fluents, then E is said to be an* explanatory set *for the human model $\mathcal{M}_H$ and a foil set F if*

$$\forall \pi \in F, \pi(I_{\gamma_E(\mathcal{M}_H)}) \not\models_{\gamma_E(\mathcal{M}_H)} G_{\gamma_E(\mathcal{M}_H)}$$

Where $\gamma_E(\mathcal{M}_H)$ is the model obtained by applying the concretizations corresponding to E on the model $\mathcal{M}_H$.

In the case of projection based abstractions of the form defined in Section 2.1, we can directly provide the model components covered by the explanatory fluent set as part of the final explanatory message provided to the user. For other abstraction techniques, we may need to employ more specialized methods

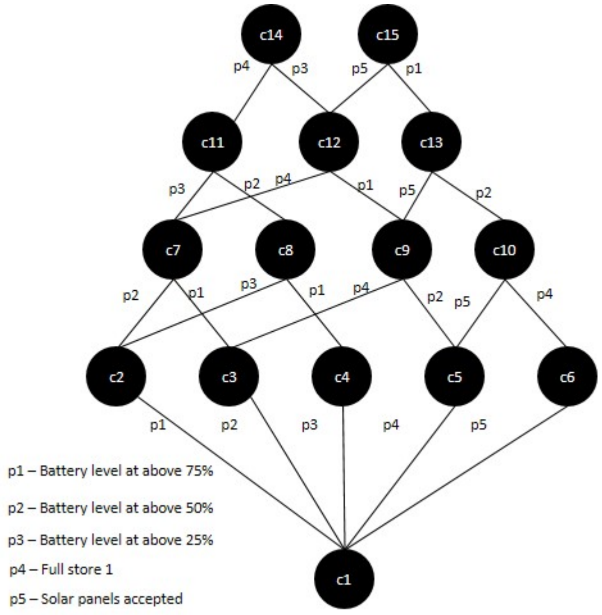

c14
c15
p4
p3
p5
p1
c11
c12
c13
p3
p2
p4
p1
p5
p2
c7
c8
c9
c10
p2
p1
p3
p1
p4
p2
p5
p4
c2
c3
c4
c5
c6
p1
p2
p3
p4
p5
p1 – Battery level at above 75%
p2 – Battery level at above 50%
p3 – Battery level at above 25%
p4 – Full store 1
p5 – Solar panels accepted
c1

Figure 2: A possible abstraction lattice for the rover domain.

to generate explanatory messages from the fluents. In Example 1 if we are to focus on the naive foil, the rover would have difficulty coming up with a single explanation as it does not know $\mathcal{M}_H$. However, it can restrict its attention to just the models that are consistent with the foils. In this scenario, it would correspond to $\{c2, c7, c8, c11, c12, c14, c15\}$.

Now we need to find a way of generating sets of explanatory fluents given this reduced set of models.

Proposition 1. *Let $\mathcal{M}_i$ be some model in $\mathcal{L}$ such that $\mathcal{M}_H \sqsubseteq \mathcal{M}_i$. If E is explanatory for $\mathcal{M}_i$ and some foil set F, then E must also explain F for $\mathcal{M}_H$.*

This proposition directly follows from the fact that for a proposition conserving lattice $\gamma_E(\mathcal{M}_i)$ will be a logical weaker model than $\gamma_E(\mathcal{M}_H)$. Next, we will define the concept of a minimal abstracting set for a given lattice $\mathcal{L}$ and foils F

Definition 6. Given an the abstraction lattice $\mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle$ and a foil set

F, the *minimal abstracting set* $\mathbb{M}^F_{min}$ is the maximal elements of the subset of all the models that are consistent with F.

$\mathbb{M}^F_{min} = \{\mathcal{M}_i | \mathcal{M}_i$ is a maximal element of $\mathbb{M}_{sat}\}$ where $\mathbb{M}_{sat} = \{\mathcal{M} \mid \mathcal{M} \in \mathbb{M}, \forall \pi \in F(\pi(I_\mathcal{M}) \models_\mathcal{M} G_\mathcal{M})\}$

Proposition 2. *For a given model lattice $\mathcal{L}$, the minimal abstracting set $\mathbb{M}^F_{min}$ is a subset of the maximal elements of the entire abstraction lattice.*

The above property ensures that when searching for the minimal abstracting set, we do not need to test the entire set of nodes or even need to know the entire lattice. In Example 1, the minimal abstracting set for the naive foil will be $\mathbb{M}^F_{min} = \{c14, c15\}$.

If we can find an explanation that is valid for all the models in $\mathbb{M}^F_{min}$ then by Proposition 1 it must work for $\mathbb{M}_H$ as well.

Proposition 3. *For a given model lattice $\mathcal{L}$ and a set of foils F and the minimal abstraction set $\mathbb{M}^F_{min}$, there exists an explanatory fluent set E such that $\forall\ \mathcal{M}' \in \mathbb{M}^F_{min}$ and $\forall \pi \in F$,*

$$\pi(I_{\gamma_E(\mathcal{M}')}) \not\models_{\gamma_E(\mathcal{M}')} G_{\gamma_E(\mathcal{M}')}$$

It is easy to see why this property holds, as any explanation that involves concretizing all possible propositions in $\mathbb{P}$ satisfies this property.

In most cases, we would prefer to compute the minimal or cheapest explanation to communicate. If all concretizations are equally expensive to communicate to the explainee, then this would correspond to finding the explanatory fluents set with the smallest size. For the naive foil in the rover example, even if the human is unaware of multiple task details, the robot can easily resolve the explainee's doubts by just explaining the concretizations related to the proposition battery_level_above_75_perc without getting into other details. Describing the details of remaining propositions is unnecessary and in the worst case might leave the human feeling overwhelmed and confused. In this case, the explanation would just include information regarding battery levels and how to identify when the battery level is or above 75% and model updates like

sample_rock-has-precondition-battery_level_above_75_perc

sample_soil-has-precondition-battery_level_above_75_perc

...

Before delving into the optimization version of the problem, let us look at the complexity of the corresponding decision problem

Theorem 2. *Given a the set of foils F and the corresponding minimal abstraction set $\mathbb{M}^F_{min}$ for a model $\mathcal{M}$, the problem of identifying whether an explanatory fluent set of size k exists for the complete lattice (which is not given) defined over an abstraction function f is* ***NP-complete****, provided the abstract function generates planning models that belong to the class described in Section 2 in polynomial time.*

Proof (Sketch). The fact that we can test the validity of the given explanation in polynomial time (size of the explanation is guaranteed to be smaller than $|\mathbb{P}|$) shows that the problem is in **NP**. We can show **NP-completeness** by reducing the set covering problem [15] to an instance of the explanation generation problem. Let's consider a set covering problem with U as the universe set and S as the set of sub-collections. Now let us create an explanation generation problem where the set of foils F is equal to U and the propositions in the set $\mathbb{P}$ contain a proposition for each member of S. Additionally concretizing with respect to a proposition will resolve only the foils covered by its corresponding subset in S. For this setting, the $\mathbb{M}^F_{min}$ consists of a single node that contains none of the propositions (and hence all the foils hold) and the concrete model contains all of them. Now if we can come up with a set of explanatory fluents of size k in this setting, then this explanation corresponds to a set cover of size k. □

The above result considers a case where the lattice needs to be generated on the fly from the minimal abstraction set. Though there may be cases where the designer may be able to provide an explicit and smaller non-proposition conserving lattice upfront. As we will see in Section 4.1, such lattices can be used to capture the designer's knowledge about the end-users.

4. Generating Optimal Explanations

As mentioned earlier, we are interested in producing the minimal explanation. Additionally, in most domains, the cost of communicating the concretization details could vary among propositions. An explanation that involves a proposition that appears in every action definition might be harder to communicate than one that only uses a proposition that is part of the definition of a single action.

In addition to the actual size, the comprehensibility of the explanations may also depend on factors like human's mental load, the familiarity with the concepts captured by the propositions, etc.. To keep our discussions simple, we will restrict the cost of communicating an explanation to just the number of unique model updates this explanation would bring about in the human model.We will use the symbol $C^{\mathcal{E}}_{p}$ to represent the cost of communicating the changes related to the proposition p and also overload it to be applicable over sets of propositions.

Now our problem is to find the optimal explanation (represented as E_{min}) for a given set of foils F or more formally

Definition 7. *A set of fluents E is said to be the optimal explanatory fluent set for the human model $\mathcal{M}_H$ and a foil set F, if*

1. *if E is an explanatory set and*
2. *there exists no other set $\widehat{E}$, such that $\widehat{E}$ is also an explanatory set and $C^{\mathcal{E}}_{E} > C^{\mathcal{E}}_{\widehat{E}}$.*

Given the fact that the human model is not known to start with, it may appear that there is no way to generate optimal explanations for the human model directly. A possible alternative might be to try identifying the set of fluents that is optimal for the set of models that could be $\mathcal{M}_H$. Calculating such an explanation naively could be extremely expensive as identifying all possible candidates for the human model would involve testing each node in the lattice for whether its a potential candidate for the human model and then searching over the space of all explanatory fluent set to find one that is optimal for the entire set of candidate models (where the optimality for a set of models is defined to be the cheapest set of fluents that is explanatory for all the models in the set). Thankfully, the properties of the lattice allow us to compute optimal solutions without keeping track of the entire set. Moreover, for lattices containing abstract models generated using procedures discussed in Section 2.1, we will see how fluent sets that are optimal for minimal abstracting set are still optimal for the original human model. *That is uncertainty over human models results in no loss of optimality.* But before proving that property, we will define the idea of the *resolution set*, that captures the specific plans resolved by concretizing the given propositions (i.e the proposition appears as an unsatisfied precondition or goal in the plan).

Definition 8. For a set of models $\mathbb{M}'$, a foil set F and a proposition p, the **resolution set** $\mathcal{R}_F(\mathbb{M}', p)$ gives the subset of foils that no longer holds in the concretized models generated through f_Λ^{safe}, i.e

$$\begin{aligned}\mathcal{R}_F(\mathbb{M}', p) = \{\pi | \pi \in F \wedge (\forall \mathcal{M}' \in \mathbb{M}'(\pi(I_{\gamma_p(\mathcal{M}')}) \not\models_{\gamma_p(\mathcal{M}')} \; G_{\gamma_p(\mathcal{M}')} \wedge \\ \pi(I_{\mathcal{M}'}) \models_{\mathcal{M}'} \; G_{\mathcal{M}'}))\}\end{aligned}$$

The idea of generating resolution sets are again closely related to the idea of resolving counter-examples used in CEGAR based method. We will also use $\mathcal{R}_F$ to also represent the set of foils resolved by a set of propositions. For notational convenience, we will use $\mathcal{R}_F(\mathbb{M}', \{\})$ to capture the subset of foils that do not hold in the current model set $\mathbb{M}'$.

Proposition 4. *For a set of model* $\mathbb{M}'$ *and a foil set* F

$$\mathcal{R}_F(\mathbb{M}', \{p_1, p_2\}) = \mathcal{R}_F(\mathbb{M}', \{p_1\}) \; \cup \; \mathcal{R}_F(\mathbb{M}', \{p_2\})$$

The above property implies that concretizing any n propositional fluents cannot resolve foils that weren't resolved by the individual fluents. The above property follows from the fact that adding a proposition into the model only resolves a foil if it adds a precondition not supported by previous actions in the plan. Since this is independent of other fluents already part of the abstraction, we can see that a set of fluents will only resolve the foils that are resolved by the individual elements of that set.

Proposition 5. *For two models $\mathcal{M}_1$, $\mathcal{M}_2$ and a set of foils F, if $\mathcal{M}_1 = f_{\Lambda}^{safe}(\mathcal{M}_2, \{p_1, .., p_k\})$ then for any proposition p,*

$$\mathcal{R}_F(\{\mathcal{M}_1\}, \{p\} \supseteq \mathcal{R}_F(\{\mathcal{M}_2\}, \{p\}) \setminus \mathcal{R}_F(\{\mathcal{M}_2\}, \{\})$$

The proposition can be established by following the definition of resolution set and rewriting the lefthand side of the equation as

$$\mathcal{R}_F(\{\mathcal{M}_2\}, p) = \mathcal{R}_F(\{\mathcal{M}_1\}, \{p_1, .., p_k\} \cup \{p\})$$

From Proposition 4 we know

$$\begin{aligned} \mathcal{R}_F(\{\mathcal{M}_2\}, p) &= \mathcal{R}_F(\{\mathcal{M}_1\}, \langle p_1, .., p_k\rangle \cdot \langle p\rangle) \\ &= \mathcal{R}_F(\{\mathcal{M}_1\}, p) \cup \mathcal{R}_F(\{\mathcal{M}_1\}, \{p_1, .., p_k\}) \end{aligned}$$

$$\begin{aligned} \mathcal{R}_F(\{\mathcal{M}_2\}, p) &= \mathcal{R}_F(\{\mathcal{M}_1\}, \langle p_1, .., p_k\rangle \cdot \langle p\rangle) \\ &= \mathcal{R}_F(\{\mathcal{M}_1\}, p) \cup \mathcal{R}_F(\{\mathcal{M}_2\}, \{\}) \end{aligned}$$

Now removing elements $\mathcal{R}_F(\{\mathcal{M}_2\}, \{\})$ from both LHS and RHS we get

$$\mathcal{R}_F(\{\mathcal{M}_2\}, p) \setminus \mathcal{R}_F(\{\mathcal{M}_2\}, \{\}) = \mathcal{R}_F(\{\mathcal{M}_1\}, p) \setminus \mathcal{R}_F(\{\mathcal{M}_2\}, \{\})$$

Which proves our original assertion.

This proposition directly leads to the following observation.

Proposition 6. *Let $\mathbb{M}_{min}^F$ be the minimal abstracting set for a foil set F and $\mathcal{M}_H$ be the human model. if every model in $\mathbb{M}_{min}^F$ if formed from $\mathcal{M}_H$ through f_{Λ}^{safe}, then for any fluent set E_{min} that is optimal for $\mathbb{M}_{min}^F$ then E_{min} must be optimal for $\mathcal{M}_H$.*

We can show the validity of the above proposition through contradiction. To start with from the definition of foils we know, $\mathcal{R}_F(\{\mathcal{M}_H\},\{\}) = \emptyset$ and thus $\mathcal{R}_F(\mathbb{M}^F_{min},\{\}) = \emptyset$. Let us assume there exists an explanatory set F_1 that is optimal for human model but not optimal for $\mathbb{M}^F_{min}$. This could only be due to two possible reasons, i.e., F_1 is not an explanatory set for $\mathbb{M}^F_{min}$ or there exists another set F_2 that is optimal for $\mathbb{M}^F_{min}$ but not applicable for $\mathcal{M}_H$. Through, Proposition 5 we have already established that any explanatory fluent set for human model must be an explanatory set for $\mathbb{M}^F_{min}$. Similarly, from Proposition 1, we know any explanatory set applicable for an abstract model set must be applicable for the concrete model as well.

Now the question is how to exactly identify E_{min}, one possibility is to perform an A* search [16] over the space of possible fluent sets to identify E_{min}. Each search state consists of the minimal set of abstract models for the human model given the current explanation prefix. We will stop the search as soon as we find a state where the foils no longer hold for the current minimal set. In addition to the systematic search, we can see that the specifics of the setting also allows us to leverage greedy search (described in Algorithm 1). In each iteration of this search, the algorithm greedily chooses the proposition that minimizes $\frac{C_p}{|F' \cap \mathcal{R}_F(\mathbb{M}',p)|}$, where F' is the set of unresolved foils at that iteration and the search ends when all foils are resolved.

Theorem 3. *The explanatory fluent set $\widehat{E}$ generated by Algorithm 1 for a set of foils F and a lattice $\mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle$ is less than or equal to $(\ln k) * C^{\mathcal{E}}_{E_{min}}$, where $C^{\mathcal{E}}_{E_{min}}$ is the cost of an optimal explanatory fluent set and k represents the maximum number of foils that can be resolved by concretizing a single proposition, i.e, $k = \max_p |\mathcal{R}_F(\mathbb{M}_{min}, p)|$.*

Proof (Sketch). We will prove the above theorem by showing that Algorithm 1 corresponds to the greedy search algorithm for a weighted set cover problem. Consider a weighted set cover problem $\langle U, S, W \rangle$ such that the universe set $U = F$, the subcollections set S is defined as $S = \{s_p | p \in \mathbb{P}\}$ where $s_p = \mathcal{R}_F(\mathbb{M}_{min}, p)$ and the cost of each subset s_p is gives as $W(s_p) = C^{\mathcal{E}}_p$. Proposition 4 ensures

Algorithm 1 Greedy Algorithm for Generating $\widehat{E}$

```
procedure Greedy-exp-search
    Input:    $\langle F, \mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle \rangle$
    Output: Explanation $\widehat{E}$
    Procedure:
    curr_model = $\langle \mathbb{M}_{min}, F \rangle$
    $\widehat{E} = \{\}$
    $\mathbb{M}_{min}$            ←MinimalAbstractModels($\mathcal{L}, F$)
    Precompute the resolution sets $\mathcal{R}_F(\mathbb{M}_{min}, p)$ for each $p \in \mathbb{P}$
    while True do
        $\mathbb{M}', F'$ = curr_model
        if $|F'| = 0$ then return $\widehat{E}$      ▷ Return $\widehat{E}$ if all the foils are resolved
        else
            $p_{next} = \underset{p}{\text{argmin}}(\frac{C_p}{|F' \cap \mathcal{R}_F(\mathbb{M}', p)|})$
            $\mathbb{M}_{new} = \{\gamma_{p_{next}}(\mathcal{M}) | \mathcal{M} \in \mathbb{M}'\}$
            curr_model = $\langle \mathbb{M}_{new}, F \setminus \mathcal{R}_F(\mathbb{M}', p) \rangle$
            $\widehat{E} = \widehat{E} \cup p$
```

that the size of resolution set is a submodular and monotonic function. In this setting, the act of identifying a set of propositions that resolve the foil set is identical to coming up with a set cover for U in the new weighted set cover problem. Furthermore, we can show that the optimal set cover $\mathcal{C}_{opt}$ must correspond to the cheapest explanation E_{min} (We can prove this equivalence using Propositions 1,3 and 4, we are skipping the details of this proof due to space constraints). Algorithm 1 describes a greedy way of identifying the cheapest set cover for this weighted set cover problem and thus the minimal explanation for the original problem. For weighted set cover the above greedy algorithm is guaranteed to generate solutions that are at most $\ln k * W(\mathcal{C}_{opt})$ [17], where $k = \max_{s \in S} |s|$ and this approximation guarantee will hold for E_{min} as well. □

We can use this algorithm to either generate solutions and or to calculate an inadmissible heuristic for the previously mentioned A* search. For the heuristic generation, we will further simplify the calculations (specifically step 8 in Algorithm 1) by considering an over-approximation of $\mathcal{R}_F$. Instead of considering the set of all foils resolved by concretizing each proposition p, we will consider the set of foils where p appears in the precondition of one of the actions in it. This set should be a superset for $\mathcal{R}_F$ for any proposition.

Now that we have formulated the basic form of explanation for this setting, we will look at how we can relax some of the assumptions made in earlier sections and how it effects the explanation generation problem. In particular, we will look at cases where the lattices are no longer proposition conserving, the users may be raising foils that are sub-optimal as opposed to invalid and finally how to support models with noise.

4.1. Supporting Explanation Generation for Non-Proposition Conserving Lattices

Proposition conserving lattices, in particular, complete lattices provide a concise way for the problem designer to specify their knowledge about the end users. In fact, with well-defined abstraction functions, they need only specify the most concrete model and the set of most abstract models to generate the rest of the lattice. Unfortunately, there may be cases where such lattices may no longer be enough to capture all information the system designer may be capable of providing about the end users. For example, consider a scenario where a robot needs to put away groceries. The goal of the robot here is to put away a set of items in prespecified storage locations. In this case, medicines need to be put in the medicine cabinet while condiments should be placed in kitchen shelves. In addition to these task-level constraints, the robot's operations are restricted by various motion level constraints that limit the possible physical movements that the robot can perform, including possible ways an object can be grasped and areas in the workspace it can reach. Clearly, these two types of constraints are quite different in terms of the background knowledge needed to understand them. While the task constraints correspond to some simple rules of the task that are easy to explain to a lay user, understanding the motion constraints require knowledge about robotics that is usually absent in most users. Thus there is a natural hierarchy in the concepts related to this task. One way to capture such information could be by controlling the order in which the various fluents are considered for abstraction, i.e., remove a particular set of fluents before moving to others (thereby making the lattice non-proposition conserving). This means, the easier to understand fluents would get introduced higher up in the lattice and the harder to understand fluent appear lower in the lattice closer to the concrete node. The task mentioned above is a particularly good fit for non-proposition conserving lattices because even the motion constraints could be captured at multiple conceptual levels. In general, non-proposition conserving lattices are a useful tool to use when you have settings where there are different propositions that capture the same phenomena but at varying levels of detail or focus on different aspects. For example, in the case of picking up an object, one could talk about the ability to pick up the object, picking up the object by grasping a particular region and even grasping using a particular grasp point on the object. We can organize the lattice in such a way that the propositions are visited in the order that reflects the preferences of the end-user. For example, for this scenario, we can arrange the concepts in such a way that simpler concepts (for example propositions related to simple reachability) are tested before moving onto more complex concepts.

While there are reasons to choose non-proposition conserving lattices and we could generate explanations using such lattices with some minor modifications on the solution method described before, the use of such lattices also have a few disadvantages. The obvious one being that the designer now have to fully specify such lattices, also the use of such lattices prevents the use of heuristics and greedy search described in earlier sections. It should also be noted that when the foils can only be resolved by introducing fluents from lower levels then the search would still need to search through all the nodes in the above before identifying the nodes that resolve the foil. Also once such a node is identified, it won't be easy to separate the set of fluent that actually contribute to resolution from those that are redundant (particularly when there are multiple foils).

To overcome these shortcomings, we will allow designers to specify a non-proposition conserving lattice while the explanation generation algorithm itself operates on a modified proposition conserving lattice that uses an updated cost function. To achieve this, we will start by defining the concept of a well-formed lattice

Definition 9. *An abstraction lattice $\mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle$ is said to be* ***well formed****, if there exists a unique minimal node (i.e the most concrete model), thus for any model $\mathcal{M} \in \mathbb{M}$, $\mathcal{M}^{\#} \sqsubseteq \mathcal{M}$.*

Any lattice we describe hence forth, will be assumed to be well-formed unless specified otherwise. While the concept of minimum abstraction set remains the same for a non-proposition conserving lattice, analyzing the results of concretizing the human model with respect to explanatory fluents requires us to look at a new concept named a completion of a lattice.

Definition 10. *For a given well formed non-proposition conserving abstraction lattice $\mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle$, a second lattice $\hat{\mathcal{L}} = \langle \hat{\mathbb{M}}, \hat{\mathbb{E}}, \mathbb{P}, \hat{\ell} \rangle$ is said to be a com-*

pletion if $\hat{\mathcal{L}}$ is a proposition conserving lattice, such that, $\mathbb{M} \subseteq \hat{\mathbb{M}}$*,* $\mathbb{E} \subseteq \hat{\mathbb{E}}$ *and* $\ell \subseteq \hat{\ell}$

A completion is relevant in this setting, because if we allow the system to freely choose propositions for the explanatory set, the updated human model (i.e the model obtained after the explanation) may not be part of the original non-proposition conserving lattice but is guaranteed to be part of the completion. Note that completions for a non-proposition conserving lattices are not unique, but in most cases we will consider a minimal completion. We can create such a completion by starting with the given lattice and adding any missing incoming edges iteratively (introducing new models only if there exists no current nodes that correspond to the set of missing propositions expected at the source of the edge).

Definition 11. *Given a non-proposition conserving lattice* $\mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle$*, it's completion* $\hat{\mathcal{L}} = \langle \hat{\mathbb{M}}, \hat{\mathbb{E}}, \mathbb{P}, \hat{\ell} \rangle$*, the human model* $\mathcal{M}_H \in \mathbb{M}$ *and the foil set* F*, a set of propositions* $E = \{p_1, ..., p_n\}$ *is said to be a set of explanatory fluents if*

$$\forall \pi \in F, \pi(I_{\gamma_E(\mathcal{M}_H)}) \not\models_{\gamma_E(\mathcal{M}_H)} G_{\gamma_E(\mathcal{M}_H)} \text{ and } \gamma_E(\mathcal{M}_H) \in \hat{\mathbb{M}}$$

As the original human model is assumed to be part of the given lattice, it must be part of the completion as well, moreover, the relation between the min abstraction set and the human model is conserved in the completion as well. This means that any set of explanatory fluents identified by using the minimum completion of the given lattice would also be valid for the human model as well. Such a minimal completion lattice, need not be created beforehand, but could in fact be generated online when searching for the explanation. Unfortunately, directly using such a completion lattice for explanation generation (once the min abstraction set is found), would result in finding sets of propositions that ignore the information captured by the given lattice. To incorporate this information we need to not only use the completion we need to consider a new cost function $C^{\mathcal{E}}_{\mathcal{L}}$ for the explanation generation.

Proposition 7. *Given a min abstraction set* $\mathbb{M}_{min}$ *for a non-proposition con-*

serving lattice $\mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle$, we can use it's completion $\hat{\mathcal{L}} = \langle \hat{\mathbb{M}}, \hat{\mathbb{E}}, \mathbb{P}, \hat{\ell} \rangle$ to identify the explanatory fluents provided the cost of explaining a given proposition p is defined as $C^{\mathcal{E}}_{\mathcal{L}}(p) = C^{\mathcal{E}}_{p} + \underset{\mathcal{M} \in \mathbb{M}_{min}}{max} L(p, \mathcal{M})$, where $L(p, \mathcal{M})$ is a penalty, such that $L(p, \mathcal{M}) \propto C^{\mathcal{E}}_{\hat{P}}$ where $\hat{P}$ is the least costly set of propositions such that $p \in \hat{P}$ and $\gamma_{\hat{P}}(\mathcal{M}) \in \mathbb{M}$.

This new penalty term ensures that a proposition is considered for explanation only after the propositions from higher levels of the given lattice is considered. Now that we are dealing with explanations using a new proposition conserving lattice, all earlier results directly carry over including the heuristic, though the search is less efficient as calculating the cost for each node requires lookup of the given lattice. Since the proposition conserving lattice is assumed to be provided upfront, we may be able to precompute the costs.

4.2. Supporting Explanations for Sub-optimal Foils

We will now consider scenarios where the explainee raises foils that are valid but may, in fact, be costlier than the one chosen by the robot. In such scenarios, we would want the robot to explain why the current plan may be preferred, but such explanations could be complicated by the fact that the actions in the domain may have state-dependent costs, for example, the cost of picking up a light block may be lower than picking up a heavier block. Here we would again need to present the user with a set of fluents and associated action costs that allow the user to correctly evaluate their alternate plans.

To investigate this setting, we will restrict our attention to cases where each action could be associated a set of positive conditional costs. We will consider a slightly updated action definition, where each action a for a model $\mathcal{M}$ is now defined by a tuple of the form $\langle prec_a, e^+_a, e^-_a, \mathcal{C}^{\mathcal{M}}_a \rangle$, where $prec_a, e^+_a$ and e^-_a are same as before and $\mathcal{C}_a$ are the set of state dependent costs associated with the action a. $\mathcal{C}^{\mathcal{M}}_a$ is itself defined as a set of individual costs of the from $\langle \phi, c \rangle$, where ϕ is a conjunction of state literals, which when satisfied in a state causes the action a to induce a cost c (where $c \in \mathbb{R}_{\geq 0}$). Now the cost of executing the action a at state s is defined

$$\mathcal{C}^{\mathcal{M}}_a(s) = \Sigma_{\langle \phi_i, c_i \rangle \in \mathcal{C}^{\mathcal{M}}_a}(\delta(\phi_i, s, c_i))$$

Where $\delta(\phi_i, s, c_i) = c_i$ if $s \models \phi_i$ else $\delta(\phi_i, s, c_i) = 0$.

We will use the function $\mathcal{C}^{\mathcal{M}}$ to return the total cost of a plan for a given initial state, i.e, for a plan $\pi = \langle a_1, ..., a_n \rangle$ and an initial state I, $\mathcal{C}^{\mathcal{M}}(\pi, I) = \mathcal{C}^{\mathcal{M}}_{a_1}(I) + ... + \mathcal{C}^{\mathcal{M}}_{a_n}(a_{n-1}(...(a_1(I))...))$.

Following the convention set by [18], we can assert that such a domain model induces a transition system of the form $\mathcal{T} = \langle S, s_0, S_g, L, T, \mathcal{C}^{\mathcal{T}} \rangle$, which is similar to the original transition system definition except that now each transition is associated with a cost determined by both source state and action. An abstract model $\mathcal{M}'$ with a transition system $\mathcal{T}'$ for a set of propositions Λ is defined in a similar way with the cost of each transition (s, a, s') given by $\mathcal{C}^{\mathcal{T}'}(s, a, s') = min(\{\mathcal{C}^{\mathcal{T}}(\hat{s}, a, \hat{s}') \mid \hat{s}, \hat{s}' \in S \wedge f_{\Lambda}(\hat{s}) = s \wedge f_{\Lambda}(\hat{s}') = s'\})$.

We will also update the explanatory setting a bit and assume that the robot presents the user with the plan and the anticipated cost of the plan in the most concrete model (denoted as $\mathcal{C}_{\pi_R}$). The user responds by providing a foil set which they believe is less costlier than the plan in question. Here we can define a set of explanatory fluents to be

Definition 12. *A set of proposition $E = \{p_1, ..., p_n\}$ is said to be* ***explanatory fluents*** *for the human model $\mathcal{M}_H$ and a foil set F if*

$$\forall \pi \in F, \pi(I_{\gamma_E(\mathcal{M}_H)}) \not\models_{\gamma_E(\mathcal{M}_H)} G_{\gamma_E(\mathcal{M}_H)} \vee \mathcal{C}^{\mathcal{M}_H}(\pi, I_{\gamma_E(\mathcal{M}_H)}) > \mathcal{C}_{\pi_R}$$

Revisiting the abstraction lattice, given the fact that we are dealing with only positive costs, the first property we can assert is that

Proposition 8. Given two models $\mathcal{M}_1$ and $\mathcal{M}_2$, such that $\mathcal{M}_1 \sqsubseteq \mathcal{M}_2$, then for any plan π, we have $\mathcal{C}(\pi, I_{\gamma_E(\mathcal{M}_1)}) \geq \mathcal{C}(\pi, I_{\gamma_E(\mathcal{M}_2)})$

This means that once we establish that a given foil is costlier than robot plan in a model, then it holds in all models that are more concrete than that one. This insight allows us to reassert Proposition 1 for this new extended definition of explanation and by extension allows us to use the idea of the minimal abstraction set in this new setting (Proposition 2 holds here as well).

This means that we can more or less directly use the search method discussed for the in-validity case here directly. Unfortunately, in this setting *the size of resolution set is no longer sub-modular* and hence we can not leverage the greedy method discussed for the pure invalidity case.

4.3. Supporting Explanations in the Presence of Human Models with Incorrect Beliefs

An underlying assumption for most of the earlier discussion has been the fact that the user's model of the task can be represented as an abstraction of the robot model, i.e. the user model may be imprecise but not incorrect. Unfortunately, this is not an assumption that can be met in all scenarios. More often than not, the user may not only be unaware of certain facts pertaining to the task but may also hold incorrect beliefs about it. Throughout this section, we will discuss how approaches discussed in earlier sections can be used to handle such cases.

Formally, let the real (but unknown) user model be $\mathcal{M}_H$ and we assert that this model is an abstraction of some (again unknown) model $\widehat{\mathcal{M}}_R$ that is defined over the same set of fluents as $\mathcal{M}_R$, but may have errors in regards to action definitions, perceived initial and goal state. Let us assume both $\widehat{\mathcal{M}}_R$ and $\mathcal{M}_H$ belong to the same class of planning problems as defined in Section 2. Again let the set of alternate plans raised by the user be F. It is important to note that the reason the user thinks these foils are valid may no longer be just due to missing fluents, but could also be due to the user's incorrect understanding of the task. This means that foils are not an accurate way of identifying the user's level of understanding, but we can still use the foils to figure out the level of abstraction at which the foils can be refuted. Though in scenarios with such models, we have to consider a complete lattice that contains all possible fluents (i.e assuming user could be wrong about the use of any of the fluents), i.e., the lattice we will use would be $\mathcal{L} = \langle \mathbb{M}, \mathbb{E}, \mathbb{P}, \ell \rangle$, where $\mathbb{P} = P_R$ (defined using f_{Λ}^{safe}). We can now use the methods described in earlier sections to find a set of explanatory fluents $\mathcal{E}$ that can refute the given set of foils. Once the information regarding the explanatory fluents in provided to the user, irrespective of the other fluents, the user should have a correct understanding of each fluents listed in $\mathcal{E}$. Let $\mathcal{M}_H + \mathcal{E}$ be the updated human model that contains the correct information about $\mathcal{E}$. Note that even though $\mathcal{M}_H$ or $\mathcal{M}_H + \mathcal{E}$ may not be part of $\mathcal{L}$, the abstraction of this updated human model that projects out all propositions absent from $\mathcal{E}$ must be part of the lattice $\mathcal{L}$, i.e, $f_{P \setminus \mathcal{E}}((\mathcal{M}_H) + \mathcal{E}) \in \gamma_{\mathcal{E}}(\mathbb{M}_{min})$. In this scenario, $\gamma_{\mathcal{E}}(\mathbb{M}_{min})$ will be singleton set and we will represent the only element in this set as $\mathcal{M}_{min}$. As per the definition of valid explanation, we know that $\mathcal{R}_F(\mathbb{M}_{min}, \mathcal{E}) = \emptyset$ and since $\gamma_{\mathcal{E}}(\mathcal{M}_H) \sqsubseteq \gamma_{\mathcal{E}}(\mathcal{M}_{min})$ and therefore the resolution set for $\gamma_{\mathcal{E}}(\mathcal{M}_H)$ must also be empty.

5. Evaluations

5.1. Empirical Evaluations on Explanation Generation for Invalid Foils

For our empirical evaluation, we wanted to understand how effective our basic approaches were in terms of the conciseness of the explanations produced, the solution computation time and the usefulness of approximation. For the approximation, we were interested in identifying the trade-off between decrease in runtime vs. reduction in solution quality. Since both explanation for incorrect beliefs and non proposition-conserving gets compiled down to finding explanation on proposition-conserving lattices, we didn't perform separate evaluations for those methods. All three explanation methods discussed in this paper (blind, heuristic and greedy) were evaluated on five IPC benchmark domains[14]. All the experiments detailed in this section were run on an Ubuntu workstation with 64G RAM.

For each domain, we selected 30 problems from either available test sets or by using standard problem generators (the problems sizes were selected to reflect the size of previous IPC test problems). The lattice for each problem-domain pair was generated by randomly selecting 50% of domain predicates

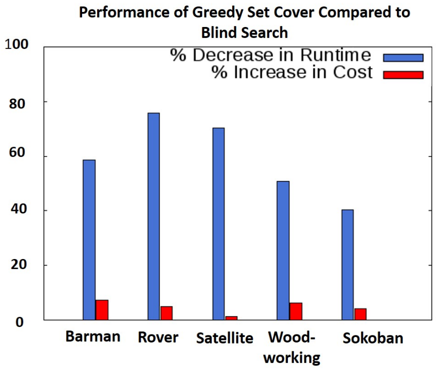


Figure 3: The graph compares the performance of greedy set cover against the optimal blind search for $|F| = 4$. It plots the average time saved by the set cover and the average increase in cost of the solution for each domain.

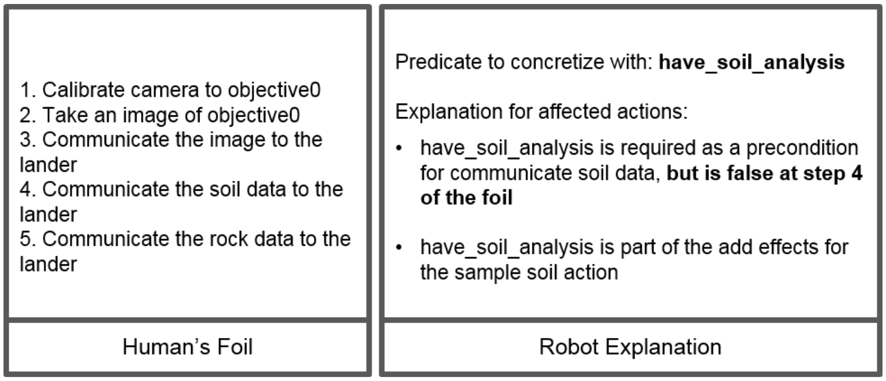
and then generating a fully connected proposition conserving lattice using that set of predicates. Since none of the models contained any conditional effects, we created the abstract models by dropping the propositions to be abstracted from the domain models (which are complete for these domains). The foils were generated by selecting random models from the lattice and creating plans from these models that do not hold in the concrete model. Each search evaluated here, generates the set of proposition whose concretizations can resolve the foils set F. In actual applications, this set of propositions needs to be converted into an explanan (the actual message) by considering how this proposition is used in the robot model. Figure 4 shows the explanation generated by our approach for a problem in Rover domain.

Table 1 presents the results from our empirical evaluation on the IPC domains. The table shows the average cost/size of each explanation along with the time taken to generate them. Note that by size, we refer to the number of predicates that are part of the explanation while the cost reflects the total number of unique model updates induced by that explanation. We attempted explanation generation for foil set sizes of one, two and four per problem.

Our main conclusion is that heuristic search seems to outperform blind

Domain Name	$C_{\mathbb{P}}$	$\lvert\mathbb{P}\rvert$	$\lvert F\rvert$	Blind Search (Optimal)			Heuristic Search			Greedy Set Cover		
				Cost	Size	Time(S)	Cost	Size	Time(S)	Cost	Size	Time(S)
Barman	84.07	7	1	6.87	1	2.43	6.87	1	2.08	6.87	1	3.61
	84	7	2	8.94	1.22	6.35	8.94	1.22	5.71	9.90	1.39	6.05
	90.7	7	4	17.19	1.77	24.99	17.19	1.77	23.7	18.45	1.97	10.34
Rover	168.66	12	1	3.58	1	7.86	3.58	1	5.22	3.58	1	19.18
	188.83	12	2	6.13	1.48	51.36	6.12	1.48	34.04	6.26	1.52	30.5
	192.83	12	4	10.87	2	203.83	10.87	2	181.87	11.42	2.19	49.32
Satellite	53.01	4	1	18.73	1	2.23	18.73	1	1.92	18.73	1	1.49
	60.77	4	2	32	1.61	7.21	32	1.6	5.86	32.53	1.7	3.04
	62.73	4	4	43.27	2.29	18.67	43.27	2.29	16.42	43.88	2.39	5.85
Woodworking	156.71	7	1	14.45	1	2.84	14.45	1	2.23	14.45	1	3.35
	146.33	7	2	20.62	1.21	6.88	20.62	1.21	4.93	21.38	1.38	6.25
	154	7	4	28.62	1.69	24.70	28.62	1.69	19.49	30.41	2	12.13
Sokoban	220.6	3	1	51.21	1	1.51	51.21	1	1.35	51.21	1	1.28
	151.72	3	2	94.52	1.55	3.93	94.52	1.55	3.35	98.31	1.73	2.59
	220.69	3	4	136.41	2.22	8.75	136.41	2.22	8.3	141.93	2.37	5.23

Table 1: Table showing runtime/cost for explanations generated for standard IPC domains. Column $|\mathbb{P}|$ represents number of predicates that were used in generating the lattice, while $C_{\mathbb{P}}^{\mathcal{E}}$ represents the cost of an explanation that tries to concretize all propositions in $\mathbb{P}$ and provides an upper bound on explanation cost.



1. Calibrate camera to objective0
2. Take an image of objective0
3. Communicate the image to the lander
4. Communicate the soil data to the lander
5. Communicate the rock data to the lander

Predicate to concretize with: **have_soil_analysis**

Explanation for affected actions:

- have_soil_analysis is required as a precondition for communicate soil data, **but is false at step 4 of the foil**
- have_soil_analysis is part of the add effects for the sample soil action

Figure 4: An example explanation generated by our system for IPC rover domain. The human incorrectly believes that the rover can communicate sample information without explicitly collecting any samples. While the abstraction lattice in this example was generated by projecting out upto 12 predicates, the search correctly identifies concretizations related to *(have_soil_analysis ?r - rover ?w - waypoint)* as the cheapest explanation ($C^{\mathcal{E}}_{E} = 2$ as opposed to $C^{\mathcal{E}}_{\mathbb{P}} = 55$)

search in almost every problem and generates near-optimal solutions (Blind search always generates the minimal explanation). Further, we saw that greedy search outperformed heuristic search in most cases barring a few exceptions. The greedy search was able to make significant gains especially for higher foil set sizes. This is entirely expected due to the fact that step 8 in Algorithm 1 can be expensive for problems with long plans (but still polynomial). This expensive pre-computation pays off as we move to cases where E_{min} consists of multiple propositions. Additionally, we found out that greedy solutions were quite comparable to the optimal solutions with respect to their costs. For example in $|F| = 4$ for satellite domain, while the greedy solution cost took a penalty of $\sim 1.4\%$ the search time was reduced by $\sim 68\%$. Figure 3 plots the comparison between the time saved by the greedy search versus any loss in optimality incurred by the greedy search.

5.2. Empirical Evaluations on Explanation Generation for Sub-optimal Foils

Next, we wanted to evaluate the empirical performance of the approach for domains with state dependent cost. For this setting, since we don't have standard benchmark domains with this property, we chose standard IPC domains and modified them to include conditional cost updates. In particular, we chose blocksworld, zenotravel, gripper and rover. For blocksworld, we introduced three new predicates, namely heavy, light and unsteady each of which takes a block as an argument. For each problem instance, we assigned each block to be either heavy or light and set some of the blocks as unsteady. We also updated the stack action so that stacking a heavy block on a light one or an already unsteady one cause the block to be unsteady. We also set a high cost penalty for stacking any block on an unsteady one. For zenotravel, we came up with three binary predicates near, farther and farthest that takes cities as arguments. We also assigned a higher cost for traveling between far away cities than nearby ones (so the optimal plan may involve the plane making a lot more stops). For gripper, we again mark a ball to be heavy or light and now each robot can also pick up two balls at the same time. We assign a high cost to picking up heavy balls and picking up the second ball in a gripper that is already holding a ball. We also provide the robot with a push action, that allows for it to move heavy balls without accruing large cost.Finally in the case of rover domains, we set some of the waypoints as being hilly area and communicating from these waypoints are assigned higher costs. Table 2 presents the explanation generation time and average explanation sizes for the modified domains. For each domain, we generated five problems and the test was run using systems of the same configuration as Section 5.1. For Blocksworld we considered instances where the number of blocks spanned from four blocks to 20, for Gripper all problems had two rooms and up to 12 balls and for Rover domain problems had upto three objectives and four waypoints. Finally, in Zenotravel all problems considered traveling between 10 cities and the number of passengers ranged from 20 to 60. The fact plan being explained were generated using optimal planners when possible and the foils were generated either using a satisficing planner (Metric-

Domain	Average Explanation Size	Runtime
BLOCKS	4.4	8.319
Gripper	5.4	7.368
Rover	4	9.690
Zeno	6.6	8.905

Table 2: The sample runtime and average explanation size for five problem instances from the modified domains.

FF [19]) or hand written using knowledge about the domain. As expected, the search was able to find the minimal number of predicates to be included into the problem to resolve the foils, for example in Blocksworld, the approach was able to correctly identify the predicate unsteady as being enough to explain the foils in the example.

In addition to the empirical results discussed in this paper on classical planning problems, the approaches discussed have also shown to be useful in modeling explanatory dialogue in the context of Task and Motion Planning (as shown in [20]).

5.3. User Study to Evaluate Role of Abstractions in Explanation

In this section, we will consider one of the assumptions that we made throughout the work, namely that providing the explanations at an abstract level would help reduce the cognitive burden on the user's end. Specifically, we will test the following hypothesis

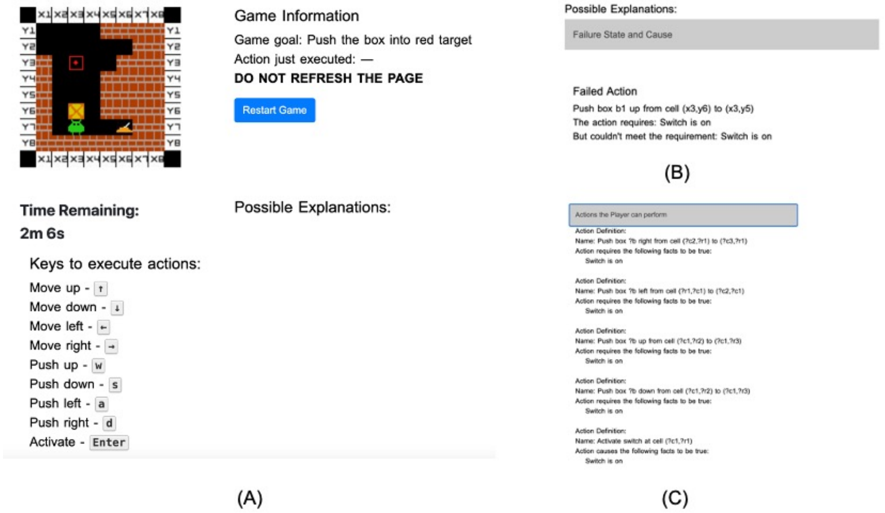
Hypothesis 1. *Given two models $\mathcal{M}_1$ and $\mathcal{M}_2$, such that $\mathcal{M}_1 \sqsubset \mathcal{M}_2$ and $\mathcal{M}_2$ is formed using methods presented in Section 2.1, a user would find it easier to work with the more abstract model $\mathcal{M}_2$ when compared to $\mathcal{M}_1$*

We will evaluate this hypothesis over two different dimensions. One with respect to the subjective workload the user may experience when working with such a model to achieve some task, and then with respect to the actual ability of the user to successfully complete the task. For the former, we will employ

NASA-Task Load Index (NASA-TLX) survey [21], while for the latter we measure the time taken by the user to complete the task. NASA-TLX is a very influential and widely used method to gauge the subjective workload experienced by the user. NASA-TLX, divides the workload of a task over six different dimensions; namely, Mental Demand, Physical Demand, Temporal Demand, Effort, Performance, and Frustration. The users are first asked to rate the task, across these dimensions on a 20 point scale (with larger value denoting higher workload). They are then required to provide relative weights across these dimensions, by making pairwise choices between these different dimensions. A weighted average of the ratings provided across these dimensions is then used as a measure of the workload.

For the actual study, we relied on a between-subject study design, wherein the study participants are divided into two groups. All study participants were students from ASU. One received abstract explanations and the other group was given concrete domain model information as explanations. As the task in question, we used a variation of the Sokoban domain, that involves the agent pushing a box to a pre-specified domain. Unlike the common versions of Sokoban, this variant involved the robot needing to first turn on a switch before pushing the boxes. Each participant in the study was allowed to play the game through a web-interface (which is shown in Figure 5, along with a sample explanation). While they were told the actions they can perform, they weren't told what each of the action achieves or their preconditions. Each player was allotted a total time of five minutes to complete the game. As the users play the game and if they perform an invalid action, they were provided with an explanation appropriate for their group.

For both groups, the current action sequence being executed was treated as the foil and the explanation consisted of the following information; the state at which the sequence failed, the specific action that failed, the expected set of preconditions, the failed precondition, and lifted model information about relevant actions. While one group of users were shown the information with respect to the concrete model, i.e., they were shown the full state, all the preconditions,



Game Information

Game goal: Push the box into red target

Action just executed: —

DO NOT REFRESH THE PAGE

Possible Explanations:

Time Remaining:
2m 6s

Keys to execute actions:

- Move up - ↑
- Move down - ↓
- Move left - ←
- Move right - →
- Push up - w
- Push down - s
- Push left - a
- Push right - d
- Activate - Enter

Possible Explanations:

Failure State and Cause

Failed Action

Push box b1 up from cell (x3,y6) to (x3,y5)
The action requires: Switch is on
But couldn't meet the requirement: Switch is on

Actions the Player can perform

Action Definition:
Name: Push box ?b right from cell (?c2,?r1) to (?c3,?r1)
Action requires the following facts to be true:
Switch is on

Action Definition:
Name: Push box ?b left from cell (?r1,?c1) to (?c2,?c1)
Action requires the following facts to be true:
Switch is on

Action Definition:
Name: Push box ?b up from cell (?c1,?r2) to (?c1,?r3)
Action requires the following facts to be true:
Switch is on

Action Definition:
Name: Push box ?b down from cell (?c1,?r2) to (?c1,?r3)
Action requires the following facts to be true:
Switch is on

Action Definition:
Name: Activate switch at cell (?c1,?r1)
Action causes the following facts to be true:
Switch is on

Figure 5: Screenshots from the user interface exposed to the end user. (A) The participant is shown the current state of the game, they are allowed to control the agent via their keyboard and whenever they perform an invalid action, they are shown a possible explanation. (B) and (C) presents a sample explanation provided to participants who were exposed to abstract explanations. Here the current state shown to the participant is empty as none of the facts are true in that state.

all failed preconditions, and the entire model of the task, the second group was shown the abstracted version of each of the above-mentioned information. The level of abstraction for this group is identified based on the foil failure. To make sure the explanation generation time is symmetric between the groups, we avoided search to identify the best level of abstraction and rather we simulated the failing foil in the most concrete model and randomly selected the predicate corresponding to one of the failing precondition to generate the abstract models. While this may result in a more detailed model than required, as we will see, even with this simple approach we did see a significant difference between the two groups. We also carried over predicates from consecutive failures, so the users of the second group saw increasingly more concrete models if they failed repeatedly (though still more abstract compared to the first group).

Scale	Concrete-Explanation Group	Abstract-Explanation Group
Effort	1.595	1.252
Frustration	2.5	2.19
Mental Demand	2.9	1.414
Performance	1.186	1.705
Physical Demand	0.038	0.152
Temporal Demand	1.929	1.719

Table 3: The weighted average workload reported by the participants of the user study across the individual scales used in NASA-TLX.

In total, we collected responses from 28 participants, 14 of whom had access to concrete explanation (henceforth referred to as Concrete-explanation group), and the remaining 14 were provided with abstract explanation (i.e the Abstract-explanation group). While the Concrete-explanation group on average took 200.857 secs to finish the task, the abstraction group only took 163.5 seconds. In terms of the weighted average workload for the Concrete-explanation group, we saw 10.147 and for the Abstract-explanation group, we found it to be 8.433. The distribution across the six scales are presented in Table 3. As seen from the table, in all but Performance and Physical demand, people reported a higher workload for the concrete explanation group. We see a particularly significant difference across the mental demand dimension, which was the main focus of the assumptions made by our work. Thus the results from both the subjective workload study, and the performance of the user (measure in terms of the time taken by the user to finish the task), conform to our original hypothesis and we see that the use of abstractions provides a distinct advantage over providing complete details.

6. Related Work

There is increasing interest within the automated planning community to solve the problem of generating explanations for plans ([22, 23]). Earlier works like [3, 4, 24] looked at explanations as a way of describing the effects of plans, while works like [25, 26] looked at plans itself as explanations for a set of observations. Another approach that has received a lot of interest recently is to view explanations as a way of achieving model reconciliation [27]. Such explanations are seen as a solution to a *model reconciliation problem* (referred to as MRP) and this approach postulates that the goal of an explanation is to update the model of the observer so they can correctly evaluate the plans in question. The methods discussed in this paper can be seen as performing a type of model-reconciliation, but one could also leverage the methods discussed here to relax some of the assumptions made by model-reconciliation works for certain conditions We discuss the relationship between model-reconciliation and the methods studied in this paper in more detail in Section 7.1

As noted, our work is closely related to the well studied method of counter-example guided refinement or CEGAR that was originally developed for Model checking [5]. Many planning works have successfully used CEGAR based methods to generate heuristics for plan generation [8, 28]. The idea of foil resolution set for a given concretization is also closely related to the process of identifying spurious counter examples employed by CEGAR based methods (cf. [29, 9, 30]). One major difference between our work and standard CEGAR based methods is the fact that in our setting the abstract model producing the foil (or counter-example) is unknown. Since we are exclusively dealing with spurious counter-examples we are also not bound to testing our foils (in other words identifying faults or pivot states) in the most concrete model (which could be quite expensive). Further, traditional CEGAR methods are generally not as focused on identifying the cheapest refinements.

Many abstraction schemes have been proposed for planning tasks (starting with [7]), but in this paper, we mainly focused on state abstractions and based our formulation on previous works like [13] and [12]. It would be interesting to see how we can extend the approaches discussed in this paper to handle temporal and procedural abstractions (e.g., HLAs [31]).

There exists a rich body of literature that has debated and discussed the role of abstraction in Social sciences (cf. [32, 33] for arguments towards abstraction, while [34] argues for adding more details provided the task constraints allow for it). Unlike these works that study explanation in everyday scenarios, explanation in the context of AI systems have a markedly different flavor, in so far that the explainer may be representing and reasoning about the task at levels of details that may be too hard for the users to understand. Thus abstraction can be a powerful tool in identifying just the required level of information to allow people to achieve their goals. This is an intuition being leveraged by more and more works to help generate explanations or even decisions that are easier to understand. For example, state abstractions have been leveraged by [35] to generate simpler models that generate easier to understand policies, and [36] uses abstraction to simplify policies. Even in the realm of machine learning explanations, abstractions have been considered as a way to generate multi-resolution explanations [37]. The importance of adjusting the level of details for different users have also been considered and argued in [38], where they propose three levels of explanations, namely, high-level, low-level, and co-created level explanations. While high and low-level explanations focus on generating summaries and detailed descriptions respectively, co-created explanations use the user interaction to determine the contents of the explanation. Our specific methods could be considered closely related to the co-created explanation studied in the paper.

There have also been recent works that have looked at generating contrastive explanations for planning. Some significant examples for these include works like [39] and [40]. Both these cases treat the cause of user's confusion to be their limited computational capabilities and the explanations tend to help them realize the consequences of following the foil without worrying about model reconciliation.

A closely related but distinct form of explanations is the one where the explanan (i.e. the information provided to the explainee) constitutes a counterfactual example [41]. Such explanations are particularly popular in classification settings, where when queried about an inexplicable classification, the system responds with a counterfactual example where the desired decision may have been made. Note that in such cases, the system needs to focus on generating counterfactual instances that the user would find acceptable. Many recent works have looked at identifying desirable properties for such counterfactual explanations (cf. [41, 42]), and some of the prominent ones identified in the literature include, making sure the counterfactual example is close enough to the decision-point in question and the counterfactual is plausible, in terms of not only being a plausible datapoint but also that it is actionable. Actionability can be particularly important in domains like loan approval, wherein the counterfactual represents the changes the user needs to make to achieve the desired outcomes. Note that in our method, it is the user who is responsible for generating the counterfactual example and as such is guaranteed to come up with foil they believed to be most likely or most useful. Thus our focus has been on ensuring that the explanations generated in response meet the desired properties discussed in the literature. As discussed above, our explanations do meet many of the important requirements discussed in the literature including being selective and social.

7. Conclusion and Discussion

In this paper, we investigated the problem of generating explanations when the explainee understands the task model at a higher level of abstraction. We looked at how we can use explanations as concretization for such scenarios and proposed algorithms for generating minimal explanations. One unique aspect of our approach is the use of foils as a way of capturing human confusion about the problem. This not only helps us formulate more efficient explanation generation methods but also aligns with the widely held belief that human expect contrastive explanations (cf. [43, 44]). Moreover, in most real-world scenarios humans usually include the foil in the request for explanations unless the foil is quite apparent from the context. The use of abstractions in explanations also allows us to reduce the cognitive burden imposed on the user for understanding the explanations. Below we have provided some more detailed discussion on the nature of explanation generated by the methods discussed in the paper and some future work.

7.1. HELM and Model Reconciliation Explanation

As mentioned earlier, the methods discussed in this case could be seen as a special case of model reconciliation [27]. Here the model updates are limited to model concretization and the human's model is an abstraction of the original model. Rather than assuming that we are given an explicit human model, we assume that the human model belongs in the set of possible models that corresponds to the various abstractions of the robot model. In this sense, this method is also comparable to the work done on generating explanations for a set of possible models [45], and in particular to the conformant explanations studied in that paper. Though unlike [45], in this setting we can guarantee that the conformant explanation is also minimal for the unknown human model (provided all model updates and hence explanations are restricted to model concretizations over fluents). Our use of minimal abstractions for explanations also allows the methods to handle cases where the user questions arise due to a mismatch in the inferential capabilities and not just a mismatch in the knowledge about the task. While the original model reconciliation work focused on explanations that address all possible foils, our work specifically tries to address foils raised by the user. This allows us to provide more concise explanations and allows us to scale to larger problems as compared to the original MRP approaches.

Another way to connect this work with model reconciliation is to leverage the insights from Section 4.3, to show that the method described in this paper can also be used in the more general model-reconciliation setting. Section 4.3 shows that for the class of planning problems studied in this paper, even when the human model may not meet the assumption that it is, in fact, an abstraction of the original robot model, we can still generate an explanation that refutes the given set of foil using abstractions formed from the robot model. Provided we use a complete abstraction lattice which contains a single maximal node formed by projecting out all the propositions. This means for explanatory queries related to just refuting alternate plans, abstraction lattices give us a way to circumvent one of the most restrictive assumptions made in model-reconciliation works, namely the need to know or learn the human model. As discussed in Section 4.3, the explanations generated over abstraction lattices will remain valid model-reconciliation explanations regardless of how the human model may be different from the robot model (provided it is still of the form described in Section 2 and doesn't contain any fluents absent from the robot model). Though compared to model-reconciliation techniques like those studied in [27, 45], the methods discussed in this paper could generate much larger explanations. For one, the explanations here involve providing information about all the uses of the explanatory fluents in the robot model, many of which the user may already know. This approach can also be extended to generate explanations of unsolvability and for partial foils of the type discussed in [46].

7.2. Properties of Explanations

The prior work on explanation as model reconciliation [27] mainly used four properties to characterize the various types of explanations that were introduced in the paper. These properties were Completeness, Monotonicity, Conciseness, and Computability. We too can use these properties to describe the explanations we have looked at (with small updates to meet our specific setting).

Any explanation generated by our methods will be complete and monotonic. While [27] defines a complete explanation as one that guarantees optimality of the plan under question. For our scenario, a complete explanation can be redefined as one that resolved all the given foils ($|\mathcal{R}_F(\Pi', E)| = |F|$). [27] considers an explanation monotonic if no future explanation can invalidate it. In our setting, this means that once a foil has been resolved by an explanation, no future explanation (or model concretizations) can reintroduce it. Which is satisfied by any explanation as concretization.

As for the remaining two properties (Conciseness and Computability), the definitions laid out in the original MRP paper directly applies for our setting as well. Similar to MRP explanations, computability and conciseness remains incompatible properties for explanations in our case too. The explanations that are easier to compute end up being neither concise nor easy to understand. For example, one simple strategy to provide explaining a plan would be to provide enough details to the explainee that the human model completely converges to the robot model, but this strategy could be extremely expensive and even unnecessary.

In addition to properties discussed in [47], works in social sciences have also prescribed some essential characteristics for what would be considered useful explanations by people [2]. Chief among them is generating contrastive explanations, which remains the central thrust of the methods discussed in this work. The other two properties usually cited by such sources are *selectiveness* and being *social*. An explanation is considered selective if it chooses to focus only on the aspects relevant to the current explanatory query. As such this is directly related to the minimality of explanation and thus the methods discussed in this paper can be considered to be selective. On the other hand, an explanation is considered social if it is tailored to the user's background. Our method supports this property in two distinct ways, one by explicitly trying to localize the user's model on the abstraction lattice, and by allowing the abstraction lattice itself to be tailored to reflect the preferences of the users.

7.3. Other Explanatory Queries

The explanatory approaches discussed in this paper have mostly focused on helping users resolve their confusion about foils, but it may also be possible that they have questions about the original robot plan. For the robot plan $\pi_R = \langle a_1, ..., a_n \rangle$, user could raise questions of the following types

1. Why perform action a_i? (where $a_i \in \pi_R$)
2. How can action a_i be performed when the precondition p is not met? (where $a_i \in \pi_R$ and $p \in \text{prec}^+_{a_i}$ or $p \in \text{prec}^-_{a_i}$ in the human model $\mathcal{M}_H$)

Question (1) captures the user's concerns regarding the use of any particular action in the plan, while question (2) captures their concerns regarding the validity of the plan. Other questions, such as achievement of goals and questions about the overall plan can be cast in terms of these more basic questions. For answering questions of the first type, we can easily adapt approaches discussed in works like [3]. For a given action, these approaches try to find causal links that capture the specific action's contributions. We can leverage the hierarchy specified by the abstraction lattice to identify causal links consisting of higher-level concepts.

For Question (2), it is possible to view such questions as another type of foil. While in earlier sections we tried to find abstract models where a particular foil can be refuted, here we just need to find the level at which the specified preconditions can be met. In the absence of disjunctive preconditions, we wouldn't need to perform a search to find such models, but rather choose the first abstract model where fluents corresponding to the preconditions in question is introduced.

This paper mostly focuses on cases where foils are fully specified. Such fully specified foils may not always be available and the human may instead be only ready to specify certain parts of the foil. In such cases, the exact foil set would consist of all plans that could potentially satisfy the plan level constraints being specified by the user question. In [46], the methods discussed in this paper have been extended to handle such cases. The work tries to handle such partial foil specifications by compiling it directly into each model in the abstraction lattice without generating the complete set of foils. Though the work only looks at employing blind search to generate such explanations. Such abstraction based explanations have also been used to generate explanations in the context of providing assistance for domain-authoring tools [48].

Acknowledgement. This research is supported in part by ONR grants N00014-16-1-2892, N00014-18-1-2442, N00014-18-1-2840, N00014-9-1-2119, AFOSR grant FA9550-18-1-0067, DARPA SAIL-ON grant W911NF-19-2-0006, NSF grants 1936997 (C-ACCEL), 1844325 and 1909370, NASA grant NNX17AD06G, and a JP Morgan AI Faculty Research grant.

References

[1] T. Chakraborti, S. Sreedharan, S. Kambhampati, The emerging landscape of explainable automated planning & decision making, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, 2020, pp. 4803–4811.

[2] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, CoRR abs/1706.07269 (2017). URL: http://arxiv.org/abs/1706.07269.

[3] B. Seegebarth, F. Müller, B. Schattenberg, S. Biundo, Making hybrid plans more clear to human users-a formal approach for generating sound explanations, in: Twenty-Second International Conference on Automated Planning and Scheduling, 2012.

[4] P. Bercher, S. Biundo, T. Geier, T. Hoernle, F. Nothdurft, F. Richter, B. Schattenberg, Plan, repair, execute, explain-how planning helps to assemble your home theater., in: ICAPS, 2014.

[5] E. Clarke, O. Grumberg, S. Jha, Y. Lu, H. Veith, Counterexample-guided abstraction refinement, in: International Conference on Computer Aided Verification, Springer, 2000, pp. 154–169.

[6] S. Sreedharan, S. Srivastava, S. Kambhampati, Hierarchical expertise-level modeling for user specific contrastive explanations, in: IJCAI, 2018.

[7] E. D. Sacerdoti, Planning in a hierarchy of abstraction spaces, Artificial intelligence 5 (1974) 115–135.

[8] J. Seipp, M. Helmert, Counterexample-guided cartesian abstraction refinement., in: ICAPS, 2013.

[9] E. R. Keyder, J. Hoffmann, P. Haslum, et al., Semi-relaxed plan heuristics., in: ICAPS, 2012.

[10] J. C. Culberson, J. Schaeffer, Pattern databases, Computational Intelligence 14 (1998) 318–334.

[11] S. Edelkamp, Planning with pattern databases, in: ECP, 2000.

[12] C. Backstrom, P. Jonsson, Bridging the gap between refinement and heuristics in abstraction, in: Twenty-Third International Joint Conference on Artificial Intelligence, 2013.

[13] S. Srivastava, S. J. Russell, A. Pinto, Metaphysics of planning domain descriptions., in: AAAI, 2016, pp. 1074–1080.

[14] International Planning Competition, IPC Competition Domains, `https://goo.gl/i35bxc`, 2011.

[15] K. Bernhard, J. Vygen, Combinatorial optimization: Theory and algorithms, Springer, Third Edition, 2005. (2008).

[16] P. E. Hart, N. J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE transactions on Systems Science and Cybernetics 4 (1968) 100–107.

[17] N. E. Young, Greedy set-cover algorithms, in: Encyclopedia of algorithms, Springer, 2008, pp. 1–99.

[18] F. Geißer, T. Keller, R. Mattmüller, Abstractions for planning with state-dependent action costs, ICAPS (2016).

[19] J. Hoffmann, The metric-ff planning system: Translating“ignoring delete lists”to numeric state variables, Journal of artificial intelligence research 20 (2003) 291–341.

[20] S. Sreedharan, M. P. Madhusoodanan, S. Srivastava, S. Kambhampati, Plan Explanation Through Search in an Abstract Model Space: Extended Results, in: ICAPS XAIP Workshop, 2018.

[21] S. G. Hart, L. E. Staveland, Development of nasa-tlx (task load index): Results of empirical and theoretical research, in: Advances in psychology, volume 52, Elsevier, 1988, pp. 139–183.

[22] M. Fox, D. Long, D. Magazzeni, Explainable Planning, in: IJCAI XAI Workshop, 2017.

[23] P. Langley, B. Meadows, M. Sridharan, D. Choi, Explainable Agency for Intelligent Autonomous Systems, in: AAAI/IAAI, 2017.

[24] S. Kambhampati, A classification of plan modification strategies based on coverage and information requirements, in: AAAI 1990 Spring Symposium on Case Based Reasoning, Citeseer, 1990.

[25] S. Sohrabi, J. A. Baier, S. A. McIlraith, Preferred explanations: Theory and generation via planning., in: AAAI, 2011.

[26] B. L. Meadows, P. Langley, M. J. Emery, Seeing beyond shadows: Incremental abductive reasoning for plan understanding., in: AAAI Workshop: Plan, Activity, and Intent Recognition, volume 13, 2013, p. 13.

[27] T. Chakraborti, S. Sreedharan, Y. Zhang, S. Kambhampati, Plan explanations as model reconciliation: Moving beyond explanation as soliloquy, in: IJCAI, 2017.

[28] J. Seipp, M. Helmert, Diverse and additive cartesian abstraction heuristics., in: ICAPS, 2014.

[29] P. Haslum, J. Slaney, S. Thiébaux, et al., Incremental lower bounds for additive cost planning problems., in: ICAPS, volume 12, 2012, pp. 74–82.

[30] M. Steinmetz, J. Hoffmann, Towards clause-learning state space search: Learning to recognize dead-ends., in: AAAI, 2016, pp. 760–768.

[31] B. Marthi, S. J. Russell, J. A. Wolfe, Angelic semantics for high-level actions., in: ICAPS, 2007, pp. 232–239.

[32] A. Garfinkel, Forms of explanation: Rethinking the questions in social theory, Yale University Press, 1982.

[33] C. Hitchcock, J. Woodward, Explanatory generalizations, part ii: Plumbing explanatory depth, Noûs 37 (2003) 181–199.

[34] C. Bechlivanidis, D. A. Lagnado, J. C. Zemla, S. Sloman, Concreteness and abstraction in everyday explanation, Psychonomic bulletin & review 24 (2017) 1451–1464.

[35] J. Ferrer-Mestres, T. G. Dietterich, O. Buffet, I. Chadès, Solving k-mdps, in: Proceedings of the International Conference on Automated Planning and Scheduling, volume 30, 2020, pp. 110–118.

[36] N. Topin, M. Veloso, Generation of policy-level explanations for reinforcement learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 2514–2521.

[37] D. Bayani, S. Mitsch, Fanoos: Multi-resolution, multi-strength, interactive explanations for learned systems, in: IJCAI XAI Workshop, 2020.

[38] K. Martin, A. Liret, N. Wiratunga, G. Owusu, M. Kern, Developing a catalogue of explainability methods to support expert and non-expert users, in: International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer, 2019, pp. 309–324.

[39] R. Eifler, M. Cashmore, J. Hoffmann, D. Magazzeni, M. Steinmetz, A new approach to plan-space explanation: Analyzing plan-property dependencies in oversubscription planning, AAAI, 2020.

[40] M. Cashmore, A. Collins, B. Krarup, S. Krivic, D. Magazzeni, D. Smith, Towards explainable ai planning as a service, arXiv preprint arXiv:1908.05059 (2019).

[41] M. T. Keane, B. Smyth, Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable ai (xai), CBR, 2020.

[42] R. M. Byrne, Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning., in: IJCAI, 2019, pp. 6276–6282.

[43] T. Lombrozo, Explanation and abductive inference, Oxford handbook of thinking and reasoning (2012) 260–276.

[44] T. Lombrozo, The structure and function of explanations, Trends in Cognitive Sciences 10 (2006) 464 – 470.

[45] S. Sreedharan, S. Kambhampati, et al., Handling model uncertainty and multiplicity in explanations via model reconciliation, in: Proceedings of the International Conference on Automated Planning and Scheduling, volume 28, 2018.

[46] S. Sreedharan, S. Srivastava, D. Smith, S. Kambhampati, Why can't you do that hal? explaining unsolvability of planning tasks, in: Proc. IJCAI, 2019.

[47] T. Chakraborti, S. Kambhampati, M. Scheutz, Y. Zhang, AI Challenges in Human-Robot Cognitive Teaming, arXiv preprint arXiv:1707.04775 (2017).

[48] S. Sreedharan, T. Chakraborti, C. Muise, Y. Khazaeni, S. Kambhampati, D3wa+: A case study of xaip in a model acquisition task, in: Proc. ICAPS, 2020.